

The DSN Programming System

A. P. Irvine
TDA Engineering Office

This article describes the DSN Programming System and its current status. A recapitulation of the System's description is included as well as major near-term milestones, plans, and some realized benefits.

I. Introduction

The goal of the DSN Programming System is the development of a body of knowledge and practices that would result in the controlled and predictable implementation of software and of those aggregates of data manipulated by software. The system incorporates a wide range of disciplines oriented toward lowering the life-cycle cost of software.

The DSN Programming System supports the on-going activities of the implementation of software in the Deep Space Network and consequently is a support system which levies requirements and provides services for all subsystems which contain software components. A discussion of all the components of the System is presented in Ref. 1.

II. Key Characteristics

A. Major Elements

The DSN Programming System includes the following:

- (1) DSN Software Standard Practices.
- (2) DSN Standard Programming Languages.
- (3) Software Implementation Aids.
- (4) Management Aids.

B. DSN Software Standard Practices

The DSN software standard practices include the following:

- (1) Software Implementation Guidelines and Practices.
- (2) Preparation of Software Requirements Documents (SRD).
- (3) Preparation of Software Definition Documents (SDD).
- (4) Preparation of Software Specification Documents (SSD).
- (5) Preparation of Software Operators Manuals (SOM).
- (6) Preparation of Software Test and Transfer Documents (STT).
- (7) DSN Data Base Implementation and Operations Guidelines and Practices.

The Software Implementation Guidelines and Practices describe the overall Tracking and Data Acquisition (TDA) software methodology, policy, and software management plan. The other standard practices provide supporting detail as summarized below.

The Standard Practices on Software Requirements Documents and Software Definition Documents cover the early

activities of requirements identification and software architectural design. These are the conceptual phases where decisions made during these formative stages tend to have profound effects on the overall costs, commitments, and the general approach. The SDD describes the software architectural design, which permits a $\pm 10\%$ accurate estimate to be made of the cost and schedule of the remaining implementation, which is the program construction. The architectural design can, in fact, be the basis for a bid by an outside contractor for a fixed-price implementation. Timing is critical and early reviews facilitate redirection, if needed. The documents covering the preparation of SSDs, SOMs, and STTs require that they be produced concurrently with the program construction and testing activities. They document the true "as-built" and "as-tested" computer program and are used to operate and maintain the transferred (delivered) program. The standard practice on data bases provides guidelines for standardizing and implementing the operational DSN Data Base. The DSN Data Base includes the data throughout the DSN in continual use for ongoing business, operations, and commitments.

C. DSN Standard Languages

Two standard languages for the implementation of software funded by the Office of Tracking and Data Acquisition have been identified. For non-real-time DSN computer programs, the MBASICtm language has been designated as the standard language. MBASICtm, an advanced version of Dartmouth BASIC, is a high-level language that combines English and simple algebra in its syntax. The term MBASICtm is derived from management-oriented BASIC.

The MBASICtm interpreter has been implemented on the PDP-10, on the UNIVAC 1108, and a subset, DEMOMBASIC, has been implemented on the MODCOMP II. A language translator, the MBASICtm Batch Processor, is currently being designed. This batch processor will provide a facility for MBASICtm programs which have been debugged in the interpretive mode to be translated into a directly executable form and stored or executed from the MBASICtm environment. The initial implementation will take place on the UNIVAC 1108. An adjunct to the MBASICtm system, the DSN File Editor, will provide a machine-independent text and data editor for MBASICtm users.

For real-time applications, HAL/S has been designated as the standard DSN language. HAL/S is a highly efficient, block-structured, high-order language which was originally developed for the NASA Space Shuttle flight software and has subsequently been designated as a NASA Standard Language.

The HAL/S compiler, to be hosted on a Modular Computer CLASSIC 7860, is being developed for JPL by Intermetrics,

Inc., the company which originally designed and developed HAL/S for NASA. The compiler, plus code generators for the MODCOMP II and the host machine, a MODCOMP CLASSIC, will be delivered in May 1980.

D. Implementation and Management Aids

A variety of aids are being developed or investigated to augment software development efforts. Among them are:

- (1) The CRISP system (Ref. 2), a set of Control Restrictive Instructions for Structural Programming, which contains at its core a program design language CRISP. CRISP imposes a structural syntax on a design language.
 - (a) CRISPFLOW is a form of CRISP that turns CRISP documentation into structural flowcharts.
 - (b) CRISP-PDL is a program design and documentation tool that has as its output cosmetized indented listings, identifier cross-references, a tier chart, a glossary, a table of contents, stub status reports, and a statistical usage summary.
- (2) "How-To" Books. It is planned that the DSN Standard will be augmented by a series of handbooks that will explain in detail how to implement the standards. The first of these handbooks, "The Life-Cycle Cost Analysis Handbook," was published as a preliminary document in 1978. The second handbook, "Preparation Guidelines for Class B Software Specification Documents," is in its second draft. Other handbooks are planned on such subjects as:
 - (a) Work Breakdown Structures.
 - (b) The Anomaly Reporting System.
 - (c) Contracting for Software as an End Item.

III. Milestones

Major milestones accomplished since the last report (Ref. 1) on the DSN Programming System are:

- (1) Completion of DSN Data Base Implementation and Operation Guidelines
Practices published December 1978
- (2) Completion of Software Methodology Textbook, Volume II, Standards distributed January 1979

The near-term milestones associated with the DSN Programming System are:

- (1) MBASICtm Batch Compiler,
Operational October 1979
- (2) HAL/S Compiler, hosted on a CLASSIC,
Operational May 1980

IV. Plans for the Future

It is possible that the replacement of the existing UNIVAC 1108 by the Large Computer Replacement (LCR) might impact the DSN Programming System. Efforts to minimize this impact have been instituted by implementing current efforts in a portable high-order language. However, past implementations such as the MBASICtm interpreter are not in a high-order language and may have to be recoded, although not redesigned, once the LCR selection is made known. Implementation of the extensions to the MBASICtm machine-independent design will have to be considered in light of the LCR selection.

One method of minimizing the impact of a new computer is to use a high-order language which will have some guarantee of portability. The TDA, in conjunction with the Flight Projects Support Office, is investigating the selection and acquisition of a standard PASCAL, which would be available

across the spectrum of computers available at the Jet Propulsion Laboratory to ensure portability. It has been proposed that the implementation of the DSN Standard File Editor be a test case to demonstrate portability between the UNIVAC 1108 and the MODCOMP II computers.

V. Benefits

An evaluation of the methodology developed under the DSN Programming System and applied to the DSN Mark III Data Subsystems Implementation Project (MDS) was made (Ref. 3). This evaluation showed that application of the methodology resulted in the following:

- (1) A small but significant savings in manpower.
- (2) A lower anomaly rate than industry-reported averages.
- (3) Sufficient management visibility to allow corrective action to be applied early enough to ensure timely delivery.

VI. Conclusion

Data has now been collected which provides evidence that the DSN Programming System does result in some expected benefits and that the goal of achieving lowered life-cycle costs for the implementation of software is realizable.

References

1. Hodgson, W. D., "The DSN Programming System," in *The Deep Space Network Progress Report 42-41*, pp. 4-9, Jet Propulsion Laboratory, Pasadena, Calif., Oct. 15, 1977.
2. Tausworth, R. C., *Standardized Development of Computer Software, Vol. II - Standards*, Jet Propulsion Laboratory, Pasadena, Calif., Aug. 1978.
3. Irvine, A., and McKenzie, M., "Evaluation of the DSN Software Methodology," in *The Deep Space Network Progress Report 42-48*, pp. 72-81, Jet Propulsion Laboratory, Pasadena, Calif., Dec. 15, 1978.